MuzeApp

MuzeApp uygulaması 2 bölümden oluşmaktadır.

1. muzeApp.Service
2. muzeApp Senaryolar

# muzeApp.Service :

muzeApp senaryo uygulamasının bilgisayar açıldığında başlamasının sağlayan, bir neden ile çöktüğünde tekrar başlatan servis yazılımıdır.

### İlgili Komutlar:

Servisin alışmasını ve durumunu kontrol etmek için aşağıdaki kodlar kullanır. Bunları raspberry üzerinden çalışırken arka planda çalışan uygulamayı kapatmak ve açmak için kullanabilirsiniz

**sudo systemctl stop muzeApp.service**

Servisin durmasını sağlar. Servisin çalıştırdığı uygulamada servisin durmasıyla sonlanır.



**sudo systemctl status muzeApp.service**

Servisin durumunu gösterir. Çalışmakta olduğunu veya durduğunu gösterir



**sudo systemctl start muzeApp.service**

Servisin tekrar çalışmasını sağlar.



**sudo nano /etc/systemd/system/muzeApp.service**

Servisin içindeki senaryo ismini değiştirmek için kullanılan kodtur.



**journalctl -f –no-pager**

Bu kod çalışan servisin hatalarını görebilmek için kullan bir kodtur. MuzeApp olarak logda araştırma yapılır.



**sudo systemctl deamon-reload**

Bu kod ile senaryo değişiminin sisteme bildirilmesi için gerekir.

### Senaryo Değiştirme

NOT: Değiştirmeden önce senaryoları incelemek için öncesinde terminaldan aşağıdaki kodu yazabilirsiniz. Aynı zamanda bu dosyadan senaryo için özel vermek istediğiniz ayarları da yapabilirisiniz. Özelliklerin açıklamarı dosyada bulunmaktadır.

sudo nano /home/pi/Projects/MuzeApp/settings.py

Raspberry Pi enerjilendirilir ve bir klavye mouse ile bir ekrana bağlanır.

Açıldıktan sonra komut satırı, yukarıdaki bardan veya ( ctrl + t ) ile açılır.

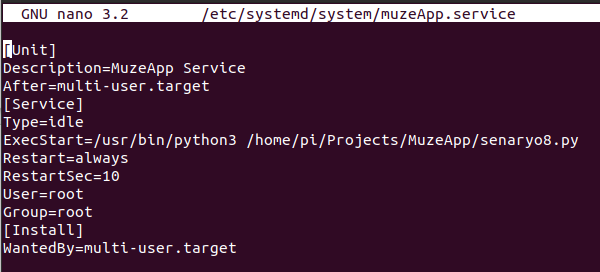
İlk önce çalışan servis durdurulur.

**sudo systemctl stop muzeApp.service**



Servisi uygulaması düzenlemek için nano ile dosya açılır

**sudo nano /etc/systemd/system/muzeApp.service**



Bu sayfada senaryo8 yazan yere istenen senaryo yazılır. Daha sonra

**(ctrl + s)** ve **(ctrl + x)**  sırasıyla basıldığında kayıt kayıt edilir. Ve şu anda tekrar komut satırında olmalısınız

Kayıdın bilgisayarA bildirilmesi için şu kod çalıştırılır

**sudo systemctl deamon-reload**



Sonrasında servis tekrar başlatılır.

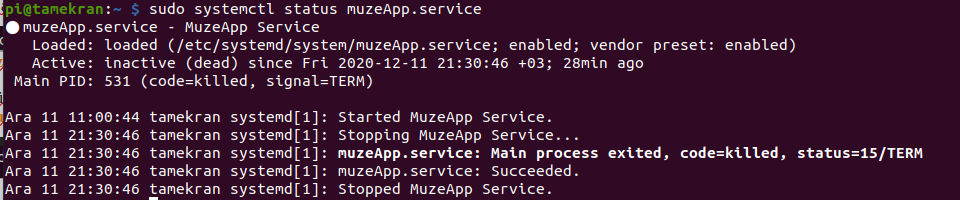


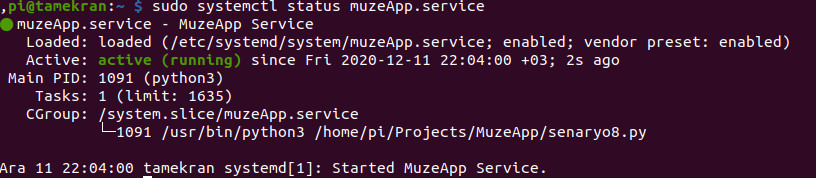
**sudo systemctl start muzeApp.service**

Servisin durumu için statüsüne bakılabilir.

**sudo systemctl status muzeApp.service**

****

Servis çalışmıyor.

Servis çalışıyor.

Servis çalıştıktan sonra raspberry reboot edilir ve tekrar başlatılır.



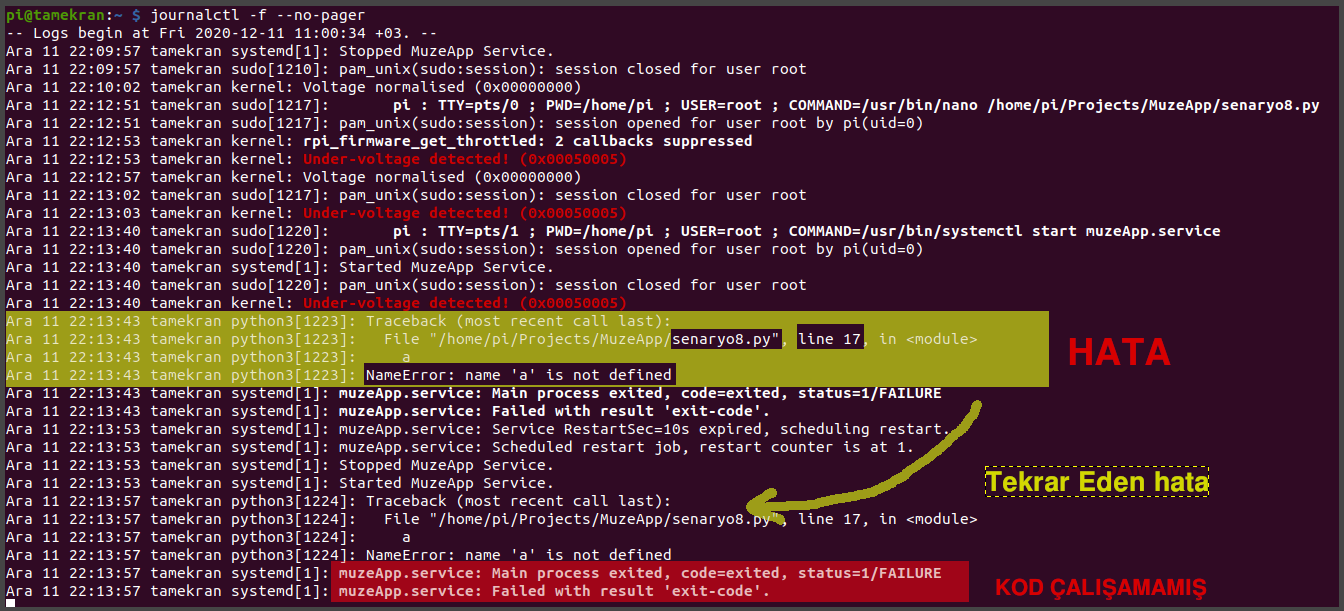
Tekrar açıldığında sistem istenildiği çalışıyorsa işlem tamamlanmıştır. (Tam olarak açılması bir kaç dk alabilir.)

Eğer sistem istenildiği çalışmıyor ise kontrol için aşağıdaki kod yazılır.



**journalctl -f –no-pager**

Bu kod servisin 10sn bir programı açmaya çalıştığı anlık olarak görülecektir. Açma sırasında bir hata oluştuysa düşecektir. Özellikle içerisinden senaryo ismi yazan hata kısmını benim ile paylaşırsanız hızlı çözüme ulaşabiliriz.

****

# muzeApp Senaryolar

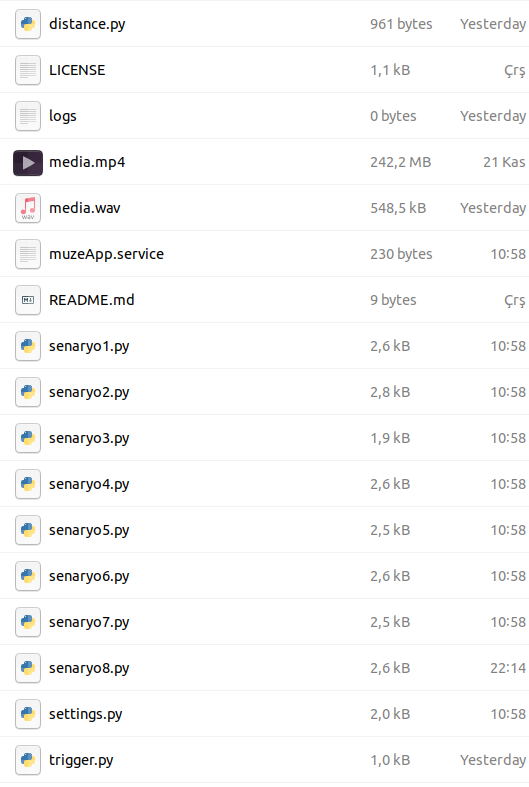
Çalışan tüm senaryolar bu proje dosyasındadır. Dosya ulaşmak için dosya yönetisinden aşağıdaki gibi klasörleri takip edebilirsiniz

**/home**

**|----/pi**

**|----/Projects**

**|----/MuzeApp/**

****

Bu klasörde medya dosyaları isim değiştirilebilir.

Settings.py dosyasında ayarlar değiştirilebilir. Detaylar aşağıdadır.

Settings.py

#### MUZE APP Interactif Video ve ses medyalarını oynama uygulaması

#### Senaryo seçimi ile farklı uygulama alanalarında kullanır.

#### Bu dosyayıdaki değişkenleri düzenledikten sonra cihazı açıp kapatılması gerekmektedir.

#### Ayarlanabilir Değişkenler

\_debug = False # True veya False yazılır. Hata ayıklamak için kullanılır

limit = 180 # Santim cinsinden, bu değer ve altı durumlarda algılar

delay = 200 # Ms cinsinden yaklaşık bir sonraki ölçüm için bekleme.

scan = 5 # Kaç ölçüm ortalaması limit değerin altındaysa sistem aktif olur

lostInTime = 5 # Saniye cinsinden algılama kaybedildikten sonra aktif kalma süresi

startDelay = 0 # Saniye cinsinden algılama oluşunca aktifleşmeden önceki bekleme

m\_width = "1920"

m\_height = "1080"

#### SENARYOLAR

#Senaryo 1

#Hareket Sensörlü Ekranlar

# Sensör algıladığında video başlar

# {lostInTime} sn eğer kimseyi göremezse video kesilir.

# Video başa döner

#Senaryo 2

#Butonlu Ekranlar

# Düğmeye basıldığında kapanana kadar video oynayacak

# {startDelay} sn sonrasında video başlayacak

#Senaryo 3

#Kulaklı Ekranlar (Videolu)

# Video sürekli oynayacak.

# Ses Kulaklıktan aktarılacak

#Senaryo 4

#Kulaklı Ekranlar (Audio)

# Düğmeye basıldığında kapana kadar ses kulaklıktan verilecek

# {startDelay} sn sonrasında ses başlayacak

#Senaryo 5

#Dev Kulaklık

# Sensör algıladığında ses başlar

# {lostInTime} sn eğer kimseyi göremezse ses kesilir.

# Video başa döner

#Senaryo 6

#Diafon

# Düğmeye basıldığında kapana kadar ses kulaklıktan verilecek

# {startDelay} sn sonrasında ses başlayacak

#Senaryo 7

#Laterna

# Sensör algıladığında ses başlar

# {lostInTime} sn eğer kimseyi göremezse ses kesilir.

# Video başa döner

#Senaryo 8

#Bavul Ekranlar

# Sensör algıladığında video başlar

# {lostInTime} sn eğer kimseyi göremezse video kesilir.

# Video başa döner